Trace Archive RFC (Proposed)

Preface

This is a revised version of the original Trace Archive RFC. The purpose of this document is to specify a means of exchange of traces and their ancillary data. In the three years that have elapsed since the Trace Archive was originally developed, the scope and uses of the data have evolved. Revisions have been made this document will clarify, and in some instances change, the contents that are to be submitted in specific fields. To accommodate new data sources, additional fields need to be added and specified. In addition, required data for specific trace types will be clarified. This proposal covers:

- Goals
- Ancillary Information
- Data Transfer
- XML
- Submission Information

The biggest changes involve:

The addition of new fields to accommodate environmental sample data.

- These additional fields allow the storage of experimental data related to the field. This is especially important as these sets of data do not fit into our normal taxonomic classification of traces.

Modification of the description of the STRATEGY and TRACE_TYPE_CODE fields.

- Currently, these fields are largely redundant. The proposal involves changing the description of the STRATEGY field to reflect the experimental strategy (rather than the sequencing strategy) used to produce a trace. This should not only better reflect the data, but also make retrieval of large sets of data easier.

Changing the SUBSPECIES_ID field to STRAIN.

- Currently, most of the data in this field is actually related to strain or cultivar information, so it makes some degree of sense to change the name of this field. Subspecies information can be related in the SPECIES_CODE field.

Addition of new requirements

- Currently, it is quite difficult to use certain data sets because there is not enough ancillary data associated with the trace. Initially, the trace archive had few restrictions because not every field is applicable to every TRACE_TYPE_CODE and STRATEGY combination. There is no change to the fields that are required for all submissions, but there now are fields that I suggested should be required for specified combinations of STRATEGY and TRACE_TYPE_CODE.

Goals

The original goal of the Trace Archive was largely related to data storage. However, over the last three years, the scope of the Trace Archive has grown substantially. The current goals of the Trace Archive are:

Archival data storage of sequence traces:

- Database storage
- Offsite storage of tapes (Coptech)

Easy retrieval of individual traces and groups of traces:

- Retrieval based on specific ancillary information
- Retrieval based on sequence comparison

The NCBI and Ensembl are collaborating to store all of the traces. There is an ongoing effort to keep the two sites synchronized with respect to trace content.

Ancillary Information

For the Trace Archive to be a useful resource, the archive must contain information describing the traces. Different sequencing centers have different naming schemes that are not mutually exclusive and it is likely that the schemes can change over time. While the name of the trace conveys some information, it is not sufficient for fully describing the data. Sequencing centers generally use the trace name as a unique key within their databases. The Trace Archive will use a combination of the center name and trace name as a unique key. In addition, every trace will be assigned a unique trace identifier. The trace identifier will be an integer value and can also function as a unique key. When the actual trace for a particular record is updated, the current TI will be replaced by a new TI. A change in TI will not occur for an update of ancillary information only. The ancillary data information should be contained in a separate file. This file can be a tab delimited text file or it can use XML format. The format of the TRACEINFO file is described in the data transfer section. The XML specification is defined in the XML section. ZTR format is defined in a separate document.

The list of ancillary data fields are described below.

Field List

The ancillary information fields defined for the Trace Archive are listed below.

RED color designates fields that are required

GREEN color designates fields that may be required, depending upon the trace type and strategy employed.

A field may be mandatory, optional or not allowed for a given combination of strategy and trace type as indicated below.

Modifications/additions from the original RFC are in red if these changes affect either a field requirement, or the definition of a field. Slight changes to affect document clarity are not noted.

***Name:* ACCESSION**
***Type:* varchar(30)**
***Example:* AC22227**
The ACCESSION is assigned upon deposition to a public repository (GenBank/EMBL/DDBJ). This field will not be applicable to all trace types (primarily WGS). However, if this field contains a valid accession identifier correlation between the primary sequence data (in Trace) and the secondary sequence data (in the public repository) is facilitated.

***Name:* AMPLIFICATION_FORWARD**
***Type:* varchar(40)**
***Example:* GGATTCTGACTAACGAGC**
The AMPLIFICATION_FORWARD field is to allow submitters to define the primers used to amplify templates for sequencing. This field is required when TRACE_TYPE_CODE=PCR or RT-PCR.

Name: **AMPLIFICATION_REVERSE**
Type: **varchar(40)**
Example: **GGATTCTGACTAACGAGC**

The AMPLIFICATION_REVERSE field is to allow submitters to define the primers used to amplify templates for sequencing. This field is required when TRACE_TYPE_CODE=PCR or RT-PCR.

Name: **AMPLIFICATION_SIZE**
Type: **int**
Example: **500**

The AMPLIFICATION_SIZE field allows submitters to define the expected amplification size for a pair of primers (defined in the AMPLIFICATION_FORWARD and AMPLIFICATION_REVERSE fields). This number should be given in base pairs. If TRACE_TYPE_CODE=PCR, the amplification size is based on amplification of genomic DNA. If the TRACE_TYPE_CODE=RT-PCR, then the amplification size is based on amplification of transcript. This field is required when TRACE_TYPE_CODE=PCR or RT-PCR.

Name: **ASSEMBLY_ID**
Type: **varchar(50)**
Example: **NCBI Build 33**

The ASSEMBLY_ID field will be a required field when the CHROMOSOME_REGION is not NULL. For groups performing re-sequencing of specific regions of a given genome, this will allow for identification of the region being re-sequenced.

Name: **BASE_FILE**
Type: **varchar(200)**
Example: **./mytraces/123clone.fasta**

Tracefiles which do not include the basecalls must provide this information in a separate file. The file designations are recorded in the BASE_FILE and QUAL_FILE fields of the TRACEINFO file. The actual bases are stored in the file designated in the BASE_FILE field. If base calls and quality scores are provided in separate files the information in these files will overwrite any information in the trace (usually *.scf) file. If the base calls and quality scores that would be provided in the BASE_FILE and QUAL_FILE are the same as the information in the trace file DO NOT PROVIDE THE FILE. Providing redundant information complicates the loading process. However, it is important to note that some formats (such as ABI) do not include the quality scores and these values must be provided as ancillary information. If the center provides the BASE_FILE and QUAL_FILE, then the peak index information should also be provided in a file called PEAK_FILE.

Name: **CENTER_NAME**
Type: **varchar(50)**
Example: **WUGSC**

Sequencing centers wishing to submit data must contact the Trace Archive administrators (trace@ncbi.nlm.nih.gov) to determine a center abbreviation. This abbreviation is used in the CENTER_NAME field. This field has a controlled vocabulary. For the complete list of submitting centers see:
http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=stat&f=xml_list_centers&m=obtain&s=center

Name: **CENTER_PROJECT**
Type: **varchar(50)**
Example: **HBBB**

The CENTER_PROJECT reflects a sequencing center's internal designation for a specific sequencing project. This field can be useful for grouping related traces.

Name: **CHEMISTRY**
Type: **varchar(50)**

Example: **BIGDYEV3.0**

Name: **CHEMISTRY_TYPE**
Type: **char(50)**
Example: **P**
The CHEMISTRY_TYPE uses a controlled list. Accepted values are:

```
Primer
Terminator
p=primer
t=terminator
```

Name: **CHROMOSOME**
Type: **varchar(8)**
Example: **11**
The CHROMOSOME indicates to which chromosome a trace has been assigned. Gene names or cytogenetic positions are not appropriate substitutes for chromosome information.

Name: **CHROMOSOME_REGION**
Type: **varchar(50)**
Example: **2:105000-106000**
This field is a required field if the ASSEMBLY_ID field is not NULL. This field can be used to describe regions that are being re-sequenced for a given genome.

Name: **CLIP_QUALITY_LEFT**
Type: **int**
Example: **56**
The CLIP_QUALITY_LEFT field indicates the base at the beginning of the sequence at which the read should be clipped due to poor quality sequence. The given value would be the first base of the high quality region of the trace.

Name: **CLIP_QUALITY_RIGHT**
Type: **int**
Example: **256**
The CLIP_QUALITY_RIGHT field indicates the base at the end of the sequence at which the read should be clipped due to poor quality sequence. The given value would be the last base of the high quality region of the trace.

Name: **CLIP_VECTOR_LEFT**
Type: **int**
Example: **75**
The CLIP_VECTOR_LEFT field indicates the base at the beginning of the sequence at which the read should be clipped due to vector sequence. The given value would be the first base of non-vector sequence.

Name: **CLIP_VECTOR_RIGHT**
Type: **int**
Example: **275**
The CLIP_VECTOR_RIGHT field indicates the base at the end of the sequence at which the read should be clipped due to vector sequence. The given value would be the last non-vector sequence.

NOTE: Many centers combine vector and quality analysis, and thus have only one set of clip values. In this case, the set of values should be placed in the CLIP_VECTOR_LEFT / CLIP_VECTOR_RIGHT fields.

NOTE: There have been some requests to make all of the clip value fields required. For various reasons, including the note above, this position has not been adopted. The decision was that either the CLIP_VECTOR_LEFT / CLIP_VECTOR_RIGHT fields should be required or the vector information (SVECTOR_ACCESSION field) should be supplied. However, since most centers use sequencing vectors that are not in GenBank/EMBL/DDBJ it seems more likely that the trim values will be given.

***Name:* CLONE_ID**
***Type:* varchar(30)**
***Example:* RP23-1123F10**

The CLONE_ID field is used to store the identifier related to an individual clone, for example a BAC clone, PAC clone or cDNA clone. If the clone is registered with the clone registry (http://www.ncbi.nlm.nih.gov/genome/clone/), standard clone registry nomenclature (see http://www.ncbi.nih.gov/genome/clone/nomenclature.shtml for more details) should be used. It is now proposed that this field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=cDNA; TRACE_TYPE_CODE=Any

STRATEGY=EST; TRACE_TYPE_CODE=Any

STRATEGY=CLONEEND; TRACE_TYPE_CODE=CLONEEND

STRATEGY=CLONE; TRACE_TYPE_CODE=Any

STRATEGY=ENCODE; TRACE_TYPE_CODE=SHOTGUN; PrimerWalk; CLONEEND

STRATEGY=FINISHING; TRACE_TYPE_CODE=Any

***Name:* CLONE_ID_LIST**
***Type:* varchar(30)**
***Example:* RP23-200A2;RP23-500P1**

The CLONE_ID_LIST field is used only if STRATEGY=PoolClone. In this case, the list of clones is provided as a semicolon delimited list. If the clones are registered with the Clone Registry, standard clone registry nomenclature should be used (see CLONE_ID field).

Note: The list of clones is not limited, but the size of the individual clone within the list is limited to 30 bytes.

It is now proposed that this field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=PoolClone; TRACE_TYPE_CODE=Any

***Name:* COLLECTION_DATE**
***Type:* datetime**
***Example:* 01/07/2003**

The COLLECTION_DATE field is used to define the date on which an environmental sample was collected. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Env Sample- Geo; TRACE_TYPE_CODE=Any

STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

***Name:* COLLECTION_TIME**
***Type:* varchar(10)**
***Example:* 07:00**

The COLLECTION_TIME field is used to define the time at which an environmental sample was collected. This field would be required for the following combination of STRATEGY and

TRACE_TYPE_CODE:
STRATEGY=Env Sample- Geo; TRACE_TYPE_CODE=Any

STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

Name: **CVECTOR_ACCESSION**
Type: **varchar(50)**
Example: **AY451994**
The CVECTOR_ACCESSION field holds the accession number for the cloning vector used. This cloning vector relates to the clone named in the CLONE_ID field.

Name: **CVECTOR_CODE**
Type: **varchar(50)**
Example: **PBACE3.6**
The CVECTOR_CODE field holds the user defined identifier for the cloning vector. Submitters are encouraged to submit all vector sequence information to public repositories. However, it is understood that many sequencing centers sequence clones from libraries they did not prepare.

Name: **DEPTH**
Type: **float**
Example: **10M**
The DEPTHfield is applicable to water samples and earth samples. If the value of this field is NULL, it is anticipated the sample was taken from the surface of the environment. While this field is only applicable to environmental samples, it is not required.

Name: **ELEVATION**
Type: **float**
Example: **500**
If the value of this field is NULL it is assumed the data were obtained at sea level. The field ELEVATION is only applicable to some environmental sample data, but is not a required field.

Name: **ENVIRONMENT_TYPE**
Type: **varchar(250)**
Example: **sea water**
The ENVIRONMENT_TYPEfield is used to describe the specific environment from which an environmental sample was taken. While the LATITUDE and LONGITUDE fields describe the location many types of environmental types could exist at this location (for example, soil, sludge, tree roots, etc). This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Env Sample- Geo; TRACE_TYPE_CODE=Any

Name: **HI_FILTER_SIZE**
Type: **varchar(50)**
Example: **50 micron**
The HI_FILTER_SIZE field is applicable only to environmental sample data but is not a required field.

Name: **HOST_CONDITION**
Type: **varchar(100)**
Example: **HIV-positive**
The HOST_CONDITION field is only applicable to environmental sample data and is used to describe the condition (healthy, sick, etc) of the host from which a sample was taken.

Name: **HOST_IDENTIFIER**
Type: **varchar(100)**
Example: **yerkes pedigree #C0479 'Clint'**

The HOST_IDENTIFIER field is only applicable to environmental sample data and is used to capture the unique name for the specific host from which a sample was obtained. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

Name: **HOST_LOCATION**
Type: **varchar(100)**
Example: **rumen**

The HOST_LOCATION field is only applicable to environmental sample data and is used to describe the specific part of the host from which the sample was obtained, for example: dental plaque, hindgut, root surfaces. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

Name: **HOST_SPECIES**
Type: **varchar(100)**
Example: **Pan troglodytes**

The HOST_SPECIES field is only applicable to environmental sample data. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

Name: **INSERT_SIZE**
Type: **int**
Example: **2000**

The INSERT_SIZE field indicates the expected insert size of the clone that is sequenced. It is understood that this is an estimate based upon the average insert sizes found in a given library. However, this information is critical for certain experiments, such as whole genome assembly. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Any; TRACE_TYPE_CODE=WGS

STRATEGY=Any; TRACE_TYPE_CODE=WCS

STRATEGY=cDNA; TRACE_TYPE_CODE=CLONEEND

STRATEGY=CLONEEND; TRACE_TYPE_CODE=CLONEEND

Name: **INDIVIDUAL_ID**
Type: **varchar(100)**
Example: **NA12345**

The INDIVIDUAL_ID field provides a center specific unique id that can associate a specific trace to an individual. This will be used primarily for population based studies (usually STRATEGY=SNP or STRATEGY=POPULATION).

Name: **INSERT_STDEV**
Type: **int**
Example: **200**

The INSERT_STDEV field reflects the approximate standard deviation of the insert size. It is understood that this information is an approximation and may change as better data is obtained. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Any; TRACE_TYPE_CODE=WGS

STRATEGY=Any; TRACE_TYPE_CODE=WCS

STRATEGY=cDNA; TRACE_TYPE_CODE=CLONEEND

STRATEGY=CLONEEND; TRACE_TYPE_CODE=CLONEEND

Name: **ITERATION**
Type: **int (1-255)**
Example: **2**

Name: **LATITUDE**
Type: **float**
Example: **54.736**
The LATITUDE field is required to describe the collection of some environmental sample data. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE: STRATEGY=Env Sample- Geo; TRACE_TYPE_CODE=Any

Name: **LO_FILTER_SIZE**
Type: **varchar(50)**
Example: **25 micron**
The LO_FILTER_SIZE field is only applicable to environmental sample data but is not a required field.

Name: **LONGITUDE**
Type: **float**
Example: **-86.403**
The LONGITUDE field is required to describe the collection of some environmental sample data. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Env Sample- Geo; TRACE_TYPE_CODE=Any

Name: **LIBRARY_ID**
Type: **varchar(30)**
Example: **RP23**
The LIBRARY_ID field documents the source library of the archival clone resource. Many genomic libraries have been registered with the Clone Registry (http://www.ncbi.nlm.nih.gov/genome/clone/) and the standard nomenclature (http://www.ncbi.nih.gov/genome/clone/nomenclature.shtml) should be used for these libraries. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=cDNA; TRACE_TYPE_CODE=Any

STRATEGY=EST; TRACE_TYPE_CODE=Any

STRATEGY=CLONEEND; TRACE_TYPE_CODE=CLONEEND

STRATEGY=CLONE; TRACE_TYPE_CODE=Any

STRATEGY=ENCODE; TRACE_TYPE_CODE=SHOTGUN; PrimerWalk; CLONEEND

STRATEGY=FINISHING; TRACE_TYPE_CODE=Any

Name: **PEAK_FILE**
Type: **varchar(200)**
Example: **./mytraces/123clone.peak**

Consult the BASE_FILE field description for more information.

Name: **PH**
Type: **float**
Example: **7.2**
The PH field is only applicable to environmental sample data but is not a required field.

Name: **PLACE_NAME**
Type: **varchar(250)**
Example: **Octopus Springs**
The PLACE_NAME field is only applicable to environmental sample data, but is not required.

Name: **PLATE_ID**
Type: **varchar(32)**
Example: **203**
The PLATE_ID and WELL_ID fields are intended to identify the storage location of the sequencing template (not the library well coordinate of an archival clone named in the CLONE_ID field). This may enable flipped or contaminated trays to be easily identified. For centers that do not use a 96 well or 384 well format for sequencing, a unique value could be used in this field (as there is now a proposal to make this a required field).

Name: **POPULATION_ID**
Type: **varchar(100)**
Example: **CEPH**
The POPULATION_ID field is used to capture center specific designations of groups of individuals. This will likely only be useful in population studies (usually STRATEGY=SNP or STRATEGY=POPULATION).

Name: **PREP_GROUP_ID**
Type: **varchar(30)**
Example: **A2**

Name: **PRIMER**
Type: **varchar(200)**
Example: **GAATACCTACGATCGCC**
The value of the PRIMER field is the actual base sequence of the sequencing primer used. If a center uses a primer extensively, the primer sequence can be entered into the list of primer codes and the PRIMER_CODE field can be used.

Name: **PRIMER_CODE**
Type: **varchar(30)**
Example: **Sp6**

Name: **PROGRAM_ID**
Type: **varchar(100)**
Example: **phred-19990722h**
The PROGRAM_ID field is used to indicate the base calling program. This field is free text. Program name, version numbers or dates are very useful. More example values:

- phred-19980904e
- abi-3.1
- ATQA
- TraceTuner

- Licor
- Megabase
- Beckman

Name: **QUAL_FILE**
Type: **varchar(200)**
Example: **./mytraces/123clone.fasta.qs**
See note associated with the BASE_FILE field.

Name: **REFERENCE_ACCESSION**
Type: **varchar(50)**
Example: **NT_029829.1**
This field is required for the following combination of STRATEGY and TRACE_TYPE_CODE: STRATEGY=Re-sequencing; TRACE_TYPE_CODE=Any

Name: **REFERENCE_OFFSET**
Type: **varchar(50)**
Example: **1520899**
This field points to the starting coordinate of the accession.version described in the REFERENCE_ACCESSION field. All coodinates should be in 1 base coordinates (i.e. sequences start at base 1, not base 0). This field is required for the following combination of STRATEGY and TRACE_TYPE_CODE: STRATEGY=Re-sequencing; TRACE_TYPE_CODE=Any

Name: **RUN_DATE**
Type: **varchar(30)**
Example: **2000-10-28**

Name: **RUN_GROUP_ID**
Type: **varchar(30)**
Example: **group2**

Name: **RUN_MACHINE_ID**
Type: **varchar(30)**
Example: **machine2**

Name: **RUN_MACHINE_TYPE**
Type: **varchar(30)**
Example: **ABI 310**

Name: **SALINITY**
Type: **float**
Example: **20%**
The SALINITY field is only applicable to environmental sample data but is not a required field.

Name: **SEQ_LIB_ID**
Type: **varchar(255)**
Example: **22194**
The SEQ_LIB_ID field is the center identifier for the M13/PUC based clone that is actually sequenced. This will allow grouping of traces by the actual ligation event and is applicable to most projects. This value will be unique within a given center. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Any; TRACE_TYPE_CODE=SHOTGUN

STRATEGY=Any; TRACE_TYPE_CODE=WGS/WCS

Name: **SOURCE_TYPE**
Type: **varchar(50)**
Example: **G (genomic DNA)**

The SOURCE_TYPE field consists of a code. Possible values are:

- G = Genomic DNA (includes PCR products from genomic DNA)
- N = Non Genomic DNA (EST, cDNA, RT-PCR, screened libraries)

Accepted values are G, N, GENOMIC, NON GENOMIC

Name: **SPECIES_CODE**
Type: **varchar(100)**
Example: **Homo sapiens**

The SPECIES_CODE field is used to classify the read by species, using proper taxonomic names where possible. This field currently is maintained as a controlled vocabulary. For a list of species currently contained within the Trace Archive, see: http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=stat&f=xml_list_species&m=obtain&s=species To submit a new species, please contact us (trace@ncbi.nlm.nih.gov) prior to submission. For cases in which it is unclear of the taxonomic origin of a specific trace (as in the case of environmental samples) the taxonomic classification 'uncultured bacteria' can be used. A second proposal for this field involves incorporating subspecies information into the species code identifier and making the field SUBSPECIES_CODE obsolete.

Name: **STRATEGY**
Type: **varchar(50)**
Example: **MODEL VERIFY**

In the original RFC, the STRATEGY field was proposed to contain the sequencing STRATEGY used in obtaining the trace. This definition made this field largely redundant with the TRACE_TYPE_CODE field. The proposal in this new version of the RFC is to make this field reflective of the experimental STRATEGY used when obtaining the trace. In some cases, this may still be redundant with the TRACE_TYPE_CODE field. Some records in the Trace Archive already contain some values in this field that are reflective of this idea. For example, the STRATEGY 'MODEL VERIFY' was proposed for a group of traces that were obtained in the process of verifying proposed gene models. In addition to conveying some information as to the original purpose of the trace, it will likely be useful in retrieving groups of traces in batch sets. It is proposed that this would be a controlled vocabulary, but that submitters would contribute to this list as needed to define various experiments and projects.

Current values:

- CCS: Concatenated cDNA sequencing
- CLONE: Clone based sequencing
- MODEL VERIFY: traces obtained to verify proposed gene models
- POOLCLONE: Pools of clones (BACs mostly)
- TRANSPOSON: Transposon based sequencing
- WCS: Whole Chromosome shotgun sequencing
- WGS: Whole Genome shotgun sequencing

Proposed values (this list would continually be expanding):

- CCS: Concatenated cDNA sequencing
- cDNA: Sequences generated in the process of sequencing cDNA clones
- CLONE: Genomic clone based (hierarchical) sequencing
- CLONEEND: Sequences generated from the end of a clone (BAC/PAC/Fosmid or cDNA)
- WGA: Whole Genome Assembly
- ENCODE: Reads generated for the Encode project
- Env Sample- GEO: Geographically generated environmental sample
- Env Sample- Host: Environmental samples collected from a specific host
- EST: single pass sequencing of cDNA templates

- FINISHING: a read specifically made for finishing, could be either BAC finishing or Whole Genome Assembly (WGA) finishing
- MODEL VERIFY: traces obtained to verify proposed gene models
- PoolClone: Pools of clones (BACs mostly)
- SNP: Reads used for SNP identification
- Re-sequencing: Re-sequencing of targeted genomic regions

Name: SUBMISSION_TYPE
Type: varchar(50)?
Example: NEW

The SUBMISSION_TYPE field allowed values:

- NEW
- UPDATE
- UPDATE INFO

Name: STRAIN
Type: varchar(50)
Example: C57BL/6J

This is a new field. Currently, many entries in the Trace Archive that are strain specific actually have the strain information in the SUBSPECIES_ID field. We should separate this information so that strain data is held in the appropriate field.

Name: SUBSPECIES_ID (to be made obsolete?)
Type: S?
Example: Verus

For many entries in the trace archive, this field actually contains strain information, rather than subspecies information. It has been proposed that this field should be made obsolete and that subspecies information should be incorporated into the SPECIES_CODE field.>

Name: SVECTOR_ACCESSION
Type: varchar(50)
Example: X52325

Name: SVECTOR_CODE
Type: varchar(50)
Example: pBluescript SK(+)

Name: TEMPERATURE
Type: float
Example: 30

The TEMPERATUREfield is only applicable to environmental sample data but it is not a required field.

Name: TEMPLATE_ID
Type: varchar(20)
Example: HBBBA2211

The TEMPLATE_ID field is used to uniquely identify the actual template that is sequenced. This field, in conjunction with the TRACE_END field, can be used to identify traces that should be marked as 'mate_pairs' because they come from opposite ends of the same clone.

Name: TRACE_DIRECTION
Type: varchar(50)

Example: **F**
The field is obsolete. Please use TRACE_END instead.

Name: **TRACE_END**
Type: **varchar(50)**
Example: **F**

The TRACE_END field can have the following values:

- **F: FORWARD**
- **R: REVERSE**
- **N: UNKNOWN**

Name: **TRACE_FILE**
Type: **varchar(200)?**
Example: **./traces/TRACE001.scf**

Name: **TRACE_FORMAT**
Type: **varchar(20)?**
Example: **scf**

The TRACE_FORMAT field can have the following values:

- **abi**
- **scf**
- **ztr**

Name: **TRACE_NAME**
Type: **varchar(250)**
Example: **HBBBA1U2211**

The TRACE_NAME field must be unique within a center, but is not required to be unique between centers. The combination of TRACE_NAME and CENTER_NAME act as a unique key within the Trace Archive.

Name: **TRACE_TYPE_CODE**
Type: **varchar(50)**
Example: **wgs**

The field TRACE_TYPE_CODE reflects the sequencing STRATEGY used to obtain the trace.

The values below are currently supported, values with * indicate this code is contained within the database:

- ***CLONEEND: BAC/PAC/fosmid end sequence**
- ***EST: Expressed sequence tag sequencing- single pass sequencing of a cDNA template**
- ***FINISHING: a read generated for finishing a BAC project**
- ***RANDOM:**
- ***RT-PCR**
- ***SHOTGUN: generally refers to BAC based shotgun sequencing**
- ***WCS: Whole Chromosome Shotgun**
- ***WGS: Whole Genome Shotgun**

Proposed values:

- **CLONEEND: Sequences generated from the end of a large insert (BAC/PAC/Fosmid) or cDNA clone**
- **EST: Single Pass Expressed Sequence Tag**
- **RT-PCR: Sequences obtained using templates generated by Reverse Transcriptase Polymerase Chain Reaction**
- **PCR: Sequences obtained using templates generated by genomic Polymerase Chain Reaction**
- **PrimerWalk: Sequences generated through a primer walking step**
- **SHOTGUN: Shotgun sequencing of clones (genomic or cDNA)**
- **TRANSPOSON: Sequences obtained using templates generated by transposons**

- WCS: Whole Chromosome Shotgun
- WGS: Whole Genome Shotgun

Name: TRANSPOSON_CODE
Type: varchar(50)
Example: Mu transposon

This TRANSPOSON_CODE field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

STRATEGY=Any; TRACE_TYPE_CODE=TRANSPOSON

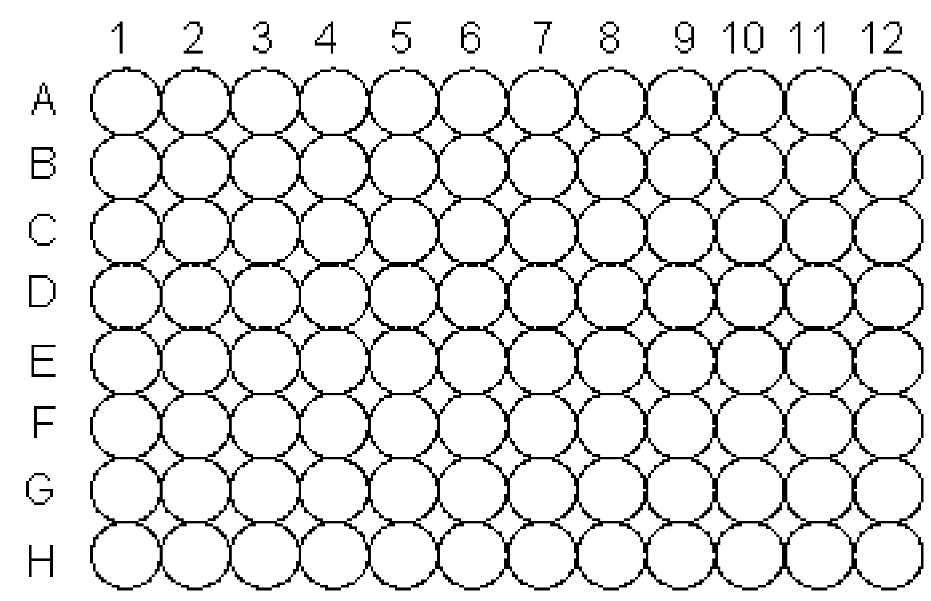
Name: TRANSPOSON_ACC
Type: varchar(50)
Example: X00913

The TRANSPOSON_ACCwould be required for the following combination of STRATEGY and TRACE_TYPE_CODE:

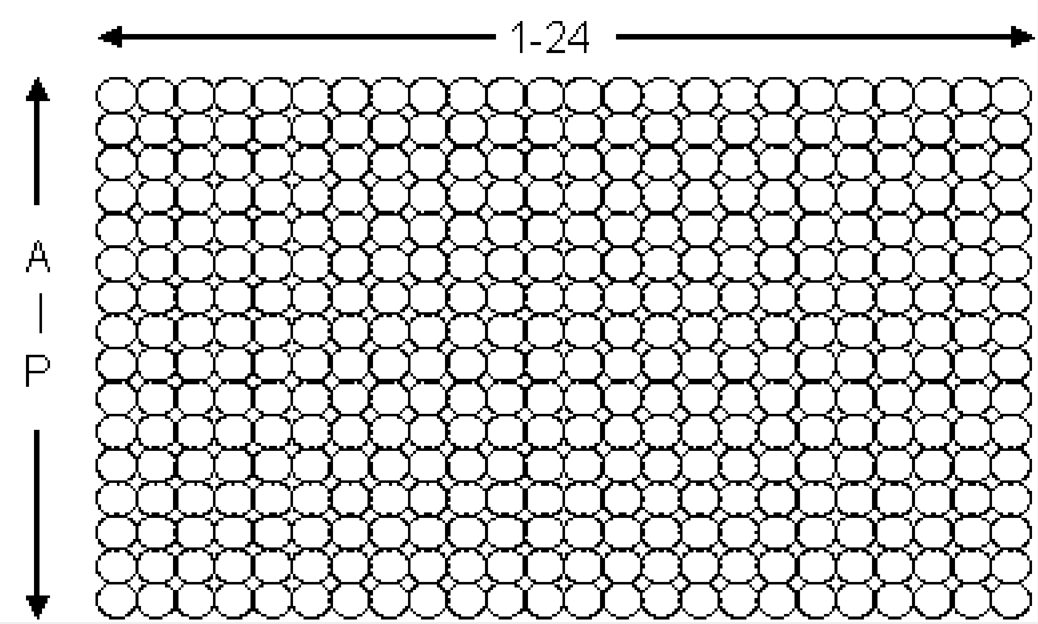
STRATEGY=Any; TRACE_TYPE_CODE=TRANSPOSON

Name: WELL_ID
Type: varchar(50)
Example: A1

The field WELL_ID in combination with the field PLATE_ID, is used to define the storage location of the sequencing reaction (see note with the field WELL_ID). Typically, sequencing reactions are performed in standard microtiter dishes having either 96 or 384 wells. (see standard configurations below).



Standard 96 well microtiter configuration



Standard 384 well microtiter configuration

Data Transfer

Given the volume of data to be moved, the method of transfer should also be specified. The unit of transfer is the trace volume. The trace volume can be provided on tapes, or placed on a secure FTP site provided by us (please contact trace@ncbi.nlm.nih.gov) to obtain a secure FTP directory. Submissions made to NCBI via ftp are automatically picked up by Ensembl. Submissions made to NCBI via tape are placed on an ftp site (after loading) for Ensembl to load. Submissions made to Ensembl are placed on NCBI ftp site to pick up and load. The volumes will have the structure below (or, they can be contained in a tar file which when extracted will have the structure below).

Care should be taken when creating large files. Many OSes and file systems (NFS) can not work with files >2 Gb. The uncompressed tar file should be > 2Gb. The filename lengths should not exceed 200 characters. Many OSes will handle >500, but shortening this length allows having parent directories when unpacking.

The volume structure should have a top directory with the same name as the volume. This top level directory is reserved for the information files describing the volume. The trace files SHOULD NOT APPEAR in the top level directory, but rather should be in a subdirectory. It is suggested that you use the name traces or the name of the project. There may be subdirectories within and this is encouraged to group traces.

The top level files are:

- README: free text describing this volume and preparation. Do not describe information which is contained in this document.
- TRACEINFO: the metadata file containing ancillary data for the field described above. This can be in XML format (TRACEINFO.xml) or tab delimited text (TRACEINFO.txt). If tab delimited the first line will have the column names.
- MD5: MD5 of all the files (MD5 tab filename).
  For example:
  ```
  728018368a7820c50cbaad633bc608a1 ./TRACEINFO
  0cbaad633bc608a1728018368a7820c5 ./traces/TRACE0001.scf
  ```

Below is an example of the volume structure:

```
./bcm-2000-07 = name of dir is same as volume in our case all of June 2000
```

```
./bcm-2000-07/README
./bcm-2000-07/MD5
./bcm-2000-07/TRACEINFO.txt
./bcm-2000-07/traces = traces are located
./bcm-2000-07/traces/HBBA = a subdir to organize the traces
./bcm-2000-07/traces/HBBA/HBBAA1U0001.scf = the first hbba trace
./bcm-2000-07/traces/HBBA/HBBAA1U0002.scf =
./bcm-2000-07/traces/HBBA/HBBAA1U0003.scf =
...lots more traces...
```

Examples are available for download: ftp://ftp.ncbi.nlm.nih.gov/pub/TraceDB/misc/examples

XML

If the trace info is provided as an XML file the info fields will serve as the tags. To preserve the grouping, the TRACE_VOLUME tag is used.

```
<?xml version="1.0"?>
<trace_volume>
<volume_name>bcm-2000-07</volume_name>
<volume_date>2000-08-11</volume_date>
<volume_version>bcm-2000-07</volume_version>
<trace>
<trace_name>HBBA0001</trace_name>
<trace_file>./traces/hbba/HBBA1U0001.scf</trace_file>
<center_name>BCM</center_name>
...more info...
</trace>
<trace>
<trace_name>HBBA0002</trace_name>
<trace_file>./traces/hbba/HBBA1U0002.scf</trace_file>
<center_name>BCM</center_name>
...more info...
</trace>
</trace_volume>
```

Submission Information

Common fields (such as CENTER_NAME) can be provided as defaults at the beginning of the submission. Example XML and Tab delimited files are available.

When a submission is loaded a log file is generated. This log file contains the ti and read name for passed reads and a list of the reads that were refected.

If greater than 5% of the reads from a particular submission fail, the entire submission will be rejected. (A submission is 1 tarball).

A tracking system has been implemented that will allow the tracking of individual submissions. Each ftp submission is given a unique tracking identifier (SID) and each tape is given a unique tape identifier (TID). Submissions can be tracked by name, ID (SID or TID), date or status. The submitting center will be notified via ftp when a submission has been processed.

After each submission has been processed log files documenting the load are placed on the ftp site.

Tape submissions should contain a label, provided by the submitting center, giving the tape a unique name. This will aid everyone in tracking data submissions.

If data are not applicable do not include the field in the submission. For example, if no chromosome information is available for a read, the CHROMOSOME field should not be included.

A read can fail for the following reasons:

- Information in the ancillary information file, but no trace file

- Zero length trace file
- Number of bases does not match the number of quality values
- There is a trace file but no ancillary information
- If the SUBMISSION_TYPE field has the value 'NEW' but the values in the CENTER_NAME and TRACE_NAME fields are already in the database, the read will be rejected.
- If the same read name is found more than one time in the tar file all reads with that name are failed.